

# Tarea Evaluable Docker - 2EV\_Ejercicio\_2\_Docker\_Compose

Módulo: Despliegue de aplicaciones Web - Distancia

Alumno: Alfonso Dapena Cores

Tarea Evaluable Docker - 2EV\_Ejercicio\_2\_Docker\_Compose

[2.1 Consideraciones previas](#)

[2.2 Descripción fichero `yaml`](#)

[2.3 Creación fichero `yaml`](#)

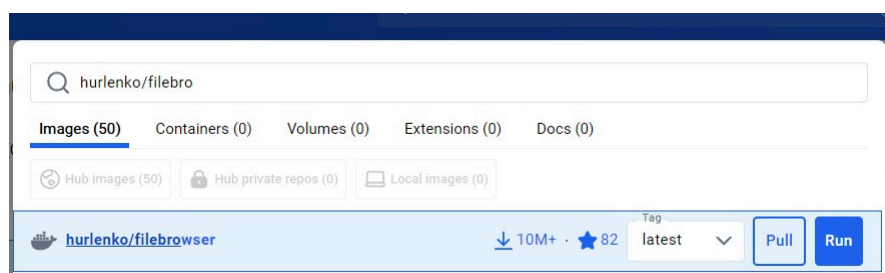
[2.4 Acceso a la aplicación](#)

## 2.1 Consideraciones previas


- Filebrowser es una herramienta de gestión de archivos accesible vía web, que permite gestionar archivos dentro de un directorio específico, permitiendo subir, eliminar, previsualizar, renombrar y editar archivos directamente desde su interfaz web, donde cada usuario tiene su propio directorio asignado. Puede funcionar como una aplicación independiente para gestionar archivos de un modo sencillo y directo.
- Docker Compose es una herramienta cuya función principal es la de simplificar la configuración y ejecución de los contenedores mediante un único archivo de configuración en formato YAML. En este archivo describiremos los servicios, redes y volúmenes necesarios para que nuestra aplicación funcione correctamente, facilitando la orquestación de contenedores. Aunque también se puede usar para crear y ejecutar un único contenedor con un único servicio, lo que ayuda a simplificar la configuración y automatización del servicio.

## 2.2 Descripción fichero `yaml`

- En primer lugar, antes de preparar el escenario donde vamos a ejecutar nuestro fichero `yaml`, debemos explorar la imagen `Filebrowser`, en nuestra aplicación Docker Desktop:



- En la documentación, hallamos un ejemplo del fichero `yaml` que vamos a ejecutar



hurlenko/filebrowser

10M+

82

View on Hub

Tag

latest

Pull

Run

Updated 2 months ago

Minimal `docker-compose.yml` may look like this:

```

version: "3"

services:
  filebrowser:
    image: hurlenko/filebrowser
    user: "${UID}:${GID}"
    ports:
      - 443:8080
    volumes:
      - /DATA_DIR:/data
      - /CONFIG_DIR:/config
    environment:
      - FB_BASEURL=/filebrowser
    restart: always

```

Simply run:

```
docker-compose up
```

- Descripción del fichero:
  1. La sección `version` es informativa y nos indica la versión con la que vamos a trabajar, en este caso la `version: "3"`, que está diseñada para ser compatible con el modo Swarm de Docker, que incluye características específicas para orquestación en clústeres.
  2. La sección `services`, especifica los servicios que vamos a crear, en este caso, crearemos el servicio al que hemos denominado `filebrowser`, que representará a un contenedor. Dentro del servicio, indicaremos la imagen en la que se va a basar nuestro contenedor: `hurlenko/filebrowser`. También definiremos el usuario `user`, y el puerto a través del que nuestra aplicación será accesible `ports`. Indicaremos, además, los volúmenes `volumes`, donde guardaremos la documentación de manera persistente, relativa a los ficheros que queremos subir a nuestra aplicación, así como la configuración del servicio. En el apartado `environment`, declararemos la `URL` para acceder a `Filebrowser`. Y, por último, en la sección `restart`, definiremos la política de reinicio: si el contenedor por cualquier motivo se para le indicaremos que intente reiniciarse cada vez que pueda.
  3. Por último, con la instrucción `docker-compose up` ( para versiones Docker Compose 1), `docker compose up` (para versiones Docker Compose 2), creamos el contenedor.

## 2.3 Creación fichero `yaml`

- En primer lugar, crearemos el directorio denominado `Filebrowser`, donde desplegaremos nuestro escenario Docker Compose. Dentro del mismo, creamos el directorio `data` donde vamos a almacenar los ficheros que vamos subir a nuestra aplicación `data`, y el directorio `config`, donde guardaremos la configuración de nuestra aplicación:

```

$ mkdir Filebrowser
$ mkdir data
$ mkdir config

```

```

adcorescliente@alfonsocliente:~$ mkdir Filebrowser
adcorescliente@alfonsocliente:~$ ls
CosasRaras  Descargas  Documentos  Filebrowser  hora.txt  Imágenes  Música  Público  Videos  webb
curl        docker.txt  Escritorio  fsSL        https:   init.sql  Plantillas  snap     web

```

```
adcorescliente@alfonsocliente:~/Filebrowser$ mkdir data
adcorescliente@alfonsocliente:~/Filebrowser$ mkdir config
adcorescliente@alfonsocliente:~/Filebrowser$ ls
config data
```

- Una vez hemos creado los directorios, debemos crear el usuario principal de la aplicación y asignarle los permisos necesarios. Después consultaremos el `uid` y el `gid` del `user` `admin`, que debemos indicar en el fichero `yaml`:

```
$ sudo useradd -m -s /bin/bash admin
$ sudo chown -R admin:admin /home/adcorescliente/Filebrowser/data
$ sudo chown -R admin:admin /home/adcorescliente/Filebrowser/config
$ id admin
```

```
adcorescliente@alfonsocliente:~$ sudo useradd -m -s /bin/bash admin
[sudo] contraseña para adcorescliente:
```

```
adcorescliente@alfonsocliente:~/Filebrowser/data$ pwd
/home/adcorescliente/Filebrowser/data
adcorescliente@alfonsocliente:~/Filebrowser/data$ sudo chown -R admin:admin /home/adcorescliente/Filebrowser/data
adcorescliente@alfonsocliente:~/Filebrowser/data$ sudo chown -R admin:admin /home/adcorescliente/Filebrowser/config
```

```
adcorescliente@alfonsocliente:~/Filebrowser/data$ id admin
uid=1001(admin) gid=1001(admin) grupos=1001(admin)
```

- Después, creamos un fichero `compose.yaml`, y con el editor `nano`, configuramos el archivo que ejecutaremos a continuación. Por último comprobaremos que nuestro contenedor está ejecutándose:

```
$ touch compose.yaml
$ nano compose.yaml
$ docker compose up -d #con esta instrucción creamos el contenedor.
$ docker ps
```

```
adcorescliente@alfonsocliente:~/Filebrowser$ touch compose.yaml
adcorescliente@alfonsocliente:~/Filebrowser$ nano compose.yaml
```

```
GNU nano 6.2 compose.yaml
version: "3"

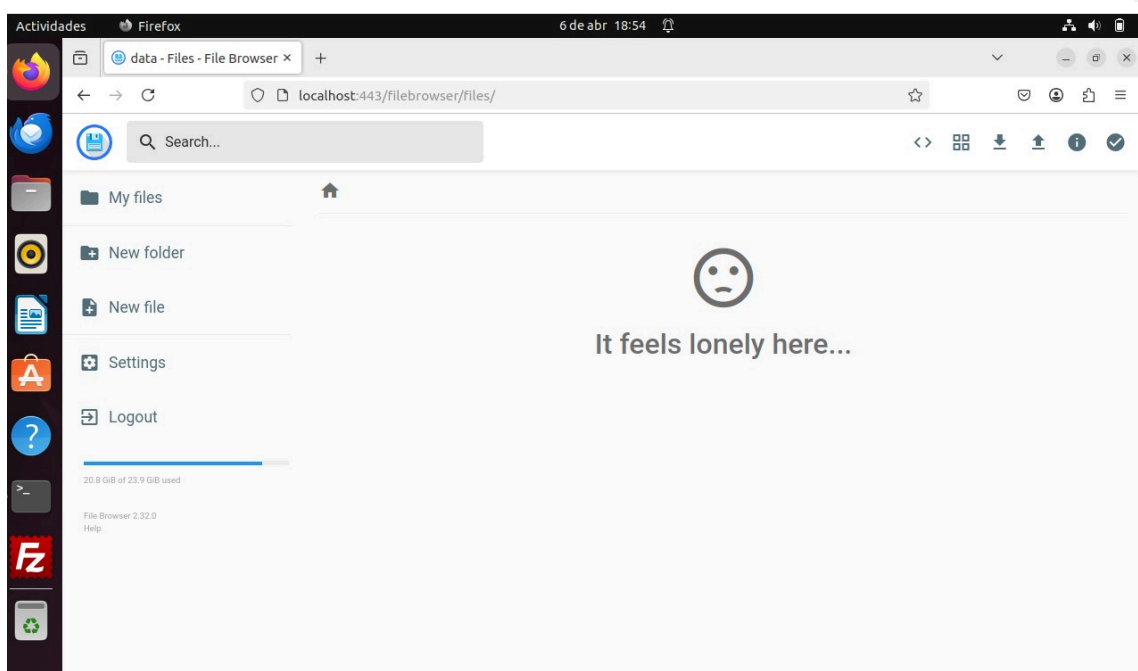
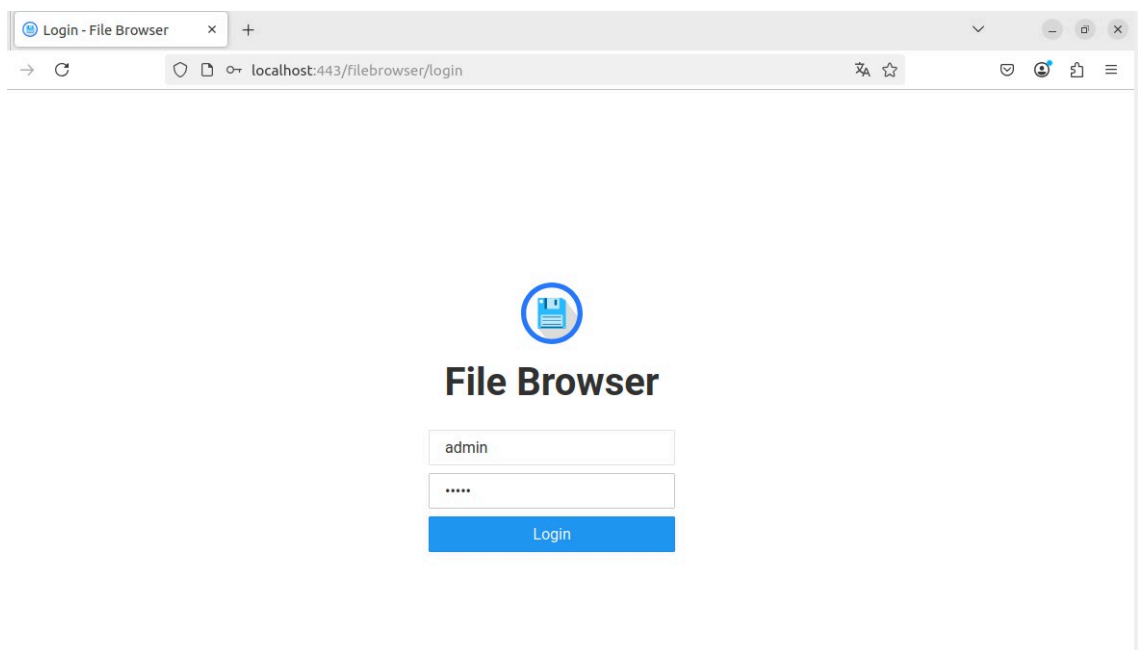
services:
  filebrowser:
    image: hurlenko/filebrowser
    user: "1001:1001"
    ports:
      - 443:8080
    volumes:
      - ./data:/data
      - ./config:/config
    environment:
      - FB_BASEURL=/filebrowser
    restart: always
```

```
adcorescliente@alfonsocliente:~/Filebrowser$ docker compose up -d
WARN[0000] /home/adcorescliente/Filebrowser/compose.yaml: the attribute 'version' is obsolete, it will be ignored, please remove it to avoid potential confusion
[+] Running 2/2
 ✓ Network filebrowser_default      Created           0.2s
 ✓ Container filebrowser-filebrowser-1 Started           0.7s

adcorescliente@alfonsocliente:~/Filebrowser$ docker ps
CONTAINER ID   IMAGE                COMMAND                  CREATED        STATUS
PORTS         NAMES
2b3d3947b602   hurlenko/filebrowser "/filebrowser --root..." 7 days ago    Up 2 hours
0.0.0.0:443->8080/tcp, [::]:443->8080/tcp   filebrowser-filebrowser-1
adcorescliente@alfonsocliente:~/Filebrowser$
```

## 2.4 Acceso a la aplicación

- A continuación, accedemos nuestro navegador e introduciremos la dirección <http://localhost:443/filebrowser>, que nos abrirá la pantalla principal de la aplicación, donde, tras logearemos, comprobaremos que todavía no hemos subido ningún fichero.



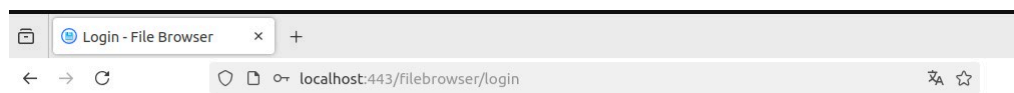
- Regresamos nuevamente a nuestro directorio `/Filebrowser/data` y creamos varios ficheros, editando el `ficheroprueba2.txt`, para añadir el texto "Hola, me llamo alfonso":

```
$ sudo touch ficheroprueba.txt
$ sudo touch fichero.txt
$ sudo touch ficheroprueba2.txt
$ sudo nano ficheroprueba2.txt
```

```
adcorescliente@alfonsocliente:~/Filebrowser/data$ sudo touch ficheroprueba2.txt
adcorescliente@alfonsocliente:~/Filebrowser/data$ sudo nano ficheroprueba2.txt
adcorescliente@alfonsocliente:~/Filebrowser/data$ ls
ficheroprueba2.txt ficheroprueba.txt fichero.txt
```



- A continuación, abrimos de nuevo el navegador e introducimos la dirección <http://localhost:443/filebrowser>, que nos lleva a la pantalla principal de la aplicación, y, tras logearnos, comprobaremos que se ha subido los ficheros correctamente a la aplicación:



## File Browser

Login

